Assignment 2: sed

CS3423 - Systems Programming Rocky Slavin - UTSA

For this assignment, you will use **sed**, **bash**, and some other command-line utilities to create a program for formatting C code. Your program should take a source code file as input and apply the following:

- No more than one space between tokens.
- No trailing whitespace after a line.
- Binary operators should always surrounded by a single space on either side (including assignment and Boolean). Only the following operators must be accounted for: `+`, `-`, `*`, `/`, `=`, `==`, `<=`, `>=`, `<`, `>`.
- Conditions should not have whitespace immediately inside of the parentheses.
- The program should *not* modify spaces which are leading, expanded tabs.
- Comments should be left alone. You may assume comments (single- and multi-line) will not appear on lines with source code.

Hint: All of the above does not need to be done in a single pass.

This assignment requires only sed and bash. **Do not** use awk, Python, or any other languages/utilities.

Example

In the code below, underscores (_) represent spaces. Note that there are no changes to comments or #include lines.

Input (inputProgram.c):

```
/**
author:_____some_student
**/
#include_<stdio.h>

int_main()_{
____int_numberIn;

____printf("Enter_a_number:_");

____scanf("%d",_&numberIn);__

____if_(_numberIn_>_10_)_{
____//____add__two
________return_numberIn_+__2;
____}_else___if____(numberIn<5){
____//____subtract_two___
________return__numberIn_-_2;

____return__numberIn*2;
}
```

Output (outputProgram.c):

```
/**
author:     some student
**/
#include <stdio.h>

int main() {
    int numberIn;

    printf("Enter a number: ");

    scanf("%d", &numberIn);

    if (numberIn > 10) {
    //    add  two
        return numberIn + 2;
    } else if (numberIn < 5){
    //    subtract two   
        return numberIn - 2;

    return numberIn * 2;
}
```

Script Execution

Your program should be invoked through a single bash file (see below) with the path to the input program as the argument. The resulting output file should be printed directly to stdout.

```
$ assign2.bash /path/to/input.txt
```

Assignment Data

A sample input file can be found in:
`/usr/local/courses/rslavin/cs3423/Fall18/assign2`.

Script Files

Your program should consist of *at least* two files:

- `assign2.bash` - the main file which is initially invoked
- *At least* one `.sed` file which is used for a sed invocation run in `assign2.bash`. Each sed invocation should have its own `.sed` file which may contain multiple sed commands.

For example, your submission may include **assign2.bash**, **command1.sed**, and **command2.sed** where the two **.sed** files are used for two corresponding sed invocations within **assign2.bash**.

Verifying Your Program

Your program must work for *arbitrary* programs by applying the rules above. You can test your program with the input provided in **inputProgram.c** and compare the output with **outputProgram.c** using **diff** (check the man-pages on how to use it).

Submission

Turn your assignment in via Blackboard. Your zip file, named **LastnameFirstname.zip** should contain only your bash and sed files.